النموذج الأول

Question 1

give the recuurance relation for this code

```
int fun2(int N)
{
  if(N<=1)
    return N;
  return fun2(N-1) + fun2(N-1); }
```

Question 2

Use master theorem to solve the following

Course Menu

$$T(n) = T(n/2) + 2^n$$

Question 3

Copy of

Use master theorem to solve the following

$$T(n) = 4T(n/2) + n^2$$

Question 4

Use master theorem to solve the following

$$T(n) = 3T(n/2) + n^2$$

Question 5

Use master theorem to solve the following

$$T(n) = 2^n T(n/2) + 2^n$$

النموذج الثاني

Question 1

0 out of 1 points

```
function printer(int n)
 for i = 1 to n do
   for j = i + 1 to n do
        print CS311
   end for
 end for
if n>0 then
for i = 1 to 4 do
printer (⌊n/2⌋)
end for
end if
```

Let T(n) denote the number of CS311 generated by a call of printer(n). i. Provide a recurrence equation for T(n).

Question 2 1 out of 1 points

```
long power(long x, long n)
if (n==0) return 1;
if (n==1) return x;
if ((n % 2) == 0)
return power(x*x, n/2);
else
return power(x*x, n/2) * x;
```

give the recurrence relation for this code

Question 3 0 out of 0.25 points

Having the following code

```
long power(long x, long n)
if (n==0) return 1;
if (n==1) return x;
if ((n % 2) == 0)
return power(power(x,n/2), 2);
else
return power(power(x,n/2), 2) *
x;
```

Do you think it will work?

answer with yes or no and state why?

النموذج الثالث

QUESTION 1

```
FIND-MAXIMUM-SUBARRAY(A, low, high)
if high == low
    return (low, high, A[low])          // base case: only one element
else mid = ⌊(low + high)/2⌋
    (left-low, left-high, left-sum) =
        FIND-MAXIMUM-SUBARRAY(A, low, mid)
    (right-low, right-high, right-sum) =
        FIND-MAXIMUM-SUBARRAY(A, mid + 1, high)
    (cross-low, cross-high, cross-sum) =
        FIND-MAX-CROSSING-SUBARRAY(A, low, mid, high)
    if left-sum ≥ right-sum and left-sum ≥ cross-sum
        return (left-low, left-high, left-sum)
    elseif right-sum ≥ left-sum and right-sum ≥ cross-sum
        return (right-low, right-high, right-sum)
    else return (cross-low, cross-high, cross-sum)
```

```
FIND-MAX-CROSSING-SUBARRAY(A, low, mid, high)
// Find a maximum subarray of the form A[i . . mid].
left-sum = −∞
sum = 0
for i = mid downto low
    sum = sum + A[i]
    if sum > left-sum
        left-sum = sum
        max-left = i
// Find a maximum subarray of the form A[mid + 1 . . j].
right-sum = −∞
sum = 0
for j = mid + 1 to high
    sum = sum + A[j]
    if sum > right-sum
        right-sum = sum
        max-right = j
// Return the indices and the sum of the two subarrays.
return (max-left, max-right, left-sum + right-sum)
```

- $T(n) = \begin{cases} 1 & if\ n = 1 \\ 2\ T(n/2) + n \end{cases}$
- $T(n) = \begin{cases} 1 & if\ n = 1 \\ 3\ T(n/2) + n \end{cases}$
- $T(n) = \begin{cases} 1 & if\ n = 1 \\ 3\ T(n/2) + n^2 \end{cases}$
- $T(n) = \begin{cases} 1 & if\ n = 1 \\ 2\ T(n/2) + n^2 \end{cases}$

1 points Save Answer

QUESTION 2

```
long power(long x, long
n)

if (n==0) return 1;
if (n==1) return x;
if ((n % 2) == 0)

return power(x,n/2) *
power(x,n/2);

else

return power(x,n.2) *
power(x,n/2) * x;
```

give the recurrence relation for this code

```
return power(x,n.2) *
power(x,n/2) * x;
```

give the recurrence relation for this code

- $T(n) = \begin{cases} C \text{ if } n = 0 \text{ or } 1 \\ 4T(n/2) + C \end{cases}$
- $T(n) = \begin{cases} C \text{ if } n = 0 \text{ or } 1 \\ 2T(n/2) + C \end{cases}$
- $T(n) = \begin{cases} C \text{ if } n = 0 \text{ or } 1 \\ 4T(n/2) + n \end{cases}$
- $T(n) = \begin{cases} C \text{ if } n = 0 \text{ or } 1 \\ 2T(n) + C \end{cases}$

QUESTION 3

```
for (i=1; i<=n*n; i++)
    for (j=0;j<i;j++<i;)
        sum++;
```

How
many times `sum++` is
executed

- ○ n^2
- ○ n^3
- ○ n^4
- ○ $n^2 \log n$

1 points Save Answer

QUESTION 4

```
int GCD(n,m){  //n>=m

if (n%m==0) return m

n=n%m

return GCD(m,n)

}
```

How many recursive calls are made by this function?

- ○ $\theta(\log n)$
- ○ $\theta(n)$
- ○ $\theta(\log \log n)$
- ○ $\theta(\sqrt{n})$